Enabling Spectrum Sharing in Secondary Market Auctions

Ian A. Kash, Rohan Murty, and David C. Parkes
Harvard University
kash@seas.harvard.edu, {rohan,parkes}@eecs.harvard.edu

Harvard University Technical Report TR-08-10

Abstract: Wireless spectrum is a scare resource, but in practice much of it is under-used by current owners. To enable better use of this spectrum, we propose an auction approach to dynamically allocate the spectrum in a secondary market. Unlike previous auction approaches, we seek to take advantage of the ability to share spectrum among some bidders while respecting the needs of others for exclusive use. Thus, unlike unlicensed spectrum (e.g. Wi-Fi), which can be shared by any device, and exclusive-use licensed spectrum, where sharing is precluded, we enable efficient allocation by supporting sharing alongside quality-of-service protections. We present SATYA (Sanskrit for "truth"), a strategyproof and scalable spectrum auction algorithm whose primary contribution is in the allocation of a right to contend for spectrum to *both* sharers and exclusive-use bidders. We demonstrate SATYA's ability to handle heterogeneous agent types involving different transmit powers and spectrum needs through extensive simulations.

1. INTRODUCTION

Spectrum is a scarce and expensive resource. For example, the 2006 Federal Communications Commission (FCC) auctions for 700 - 800 MHz are estimated to have raised almost $19 billion dollars. Hence, the barrier to entry for potential spectrum buyers is high. One can either buy a lease on spectrum covering a large area at a high price or use the limited spectral bands classified as unlicensed (e.g. Wi-Fi). Such unlicensed bands are subject to a "tragedy of the commons" where, since they are free to use, they are over-used and performance suffers [9]. Efforts such as the recent FCC ruling on white spaces are attempting to free additional spectrum by permitting opportunistic access [4]. However, such efforts are being met with opposition by incumbents (such as TV broadcasters and wireless microphones manufacturers) who have no incentive to permit their spectrum to be shared.

Motivated by these observations, many researchers and companies (*e.g.,* [7, 35, 19]) have proposed allowing spectrum owners and spectrum users to participate in a secondary market for spectrum where users are allocated the use of spectrum in a small area on a dynamic basis based on their short- or medium-term needs. This approach is beneficial for two reasons. First, it allows flexible approaches to determining how best to allocate spectrum rather than relying on the decision making of regulators like the FCC in the United States. Second, it provides an incentive for spectrum that is currently owned but unused or under-used to be made available by its owners. Note, by secondary market we mean, one in which the owner of a chunk of spectrum leases different frequencies to *other* users who bid for the spectrum. The FCC has also recognized the potential use of a secondary spectrum market and has begun encouraging spectrum owners in certain bands to sublease the spectrum. [18].

Prior work has proposed a number of auction designs to support such a market. However, the possibilities for **sharing** in such markets have not been sufficiently explored. Most auctions provide exclusive access: the allocation is such that no winners will interfere. However, this may often not be the most efficient use of spectrum. For example, devices like wireless microphones are only used occasionally, so even if they require exclusive access while in use, some other device may be able to use the same spectrum on a secondary basis when they are not. This *heterogeneity* of devices and demands is one source of opportunities for sharing. Further, many devices are capable of using a medium access controller (MAC) to share bandwidth when given the right to contend.

Rather than full sharing, as in the Wi-Fi model, using an auction has two key advantages. First, it provides revenue and thus incentives for primary spectrum owners to open up spectrum to other uses. Second, it provides incentives for different potential users to describe (through bids) their distinct needs for spectrum access, be it exclusive or with sharing. With Wi-Fi, if too many people try to use the same access point, service degrades and may become unacceptable for all of them, and no one has an incentive to consider the (negative) externality their use imposes on others.

We present SATYA, a scalable auction-based algorithm that permits different classes of spectrum users (sharing and exclusive) to co-exist and share the spectrum whenever desirable while appropriately accounting for the resulting externalities. While SATYA relies on the MAC to arbitrate access to the spectrum, the precise set of contending nodes is decided based on the outcome of the auction.

SATYA uses a simple, yet expressive, language to allow bidders to express their value for different allocations given probabilistic activation patterns, interference, and different requirements for shared vs exclusive-access spectrum. This kind of expressiveness is necessary to enable the efficient allocation of short-term spectrum rights to multiple devices. To evaluate SATYA we use real world data sources to determine participants in the auction, along with the sophisticated Longley-Rice propagation model [3] and high resolution terrain information to generate conflict graphs. We compare the performance of SATYA against other auction algorithms and baseline computations. We also demonstrate how *reserve prices* can be used to increase the revenue of spectrum auctions, an important consideration when trying to encourage spectrum owners to participate.

One particular design choice we have made with SATYA is that it is a *strategyproof* auction. Roughly speaking, this means it is optimal for a bidder to reveal his true valuation when bidding. This *strategic simplicity* is one key advantage of a strategyproof auction: devices do not need to be programmed with sophisticated strategies and employ wasteful counterspeculation, and users can do well simply expressing preferences directly and straightforwardly [32]. The other key advantage of strategyproofness is that it makes evaluating SATYA much easier. The allocative efficiency and revenue of a strategyproof auction can be evaluated by examining what happens when bidders reveal their true values. For non-strategyproof auctions, we first have to reason about how strategic bidders would behave, a much more difficult problem.

Current proposals for secondary-market spectrum auctions are not compatible with the externalities created by sharing or are not scalable. The essential difficulty is that by ignoring the possibility of sharing, they rely on bidders caring only about whether or not they are allocated a channel. With sharing, bidders also care *with whom* they will share the channel. SATYA uses a novel combination of the technical methods of "bucketing" and "ironing," which enables a tractable, strategyproof auction despite externalities. We discuss these methods further in Section 4.

In summary, this paper makes the following contributions:

- The first strategyproof, scalable auction design for dynamic spectrum access that allows *sharing* and exclusive access by appropriately dealing with the externalities this creates.
- An approach that accommodates different classes of wireless users, each with a different transmit power, spectrum access, and activation patterns.
- The use of sophisticated propagation models and real world data to demonstrate the efficacy of SATYA, including the use of reserve prices to increase the revenue from auctions in secondary markets.

2. RELATED WORK

Most spectrum auction algorithms do not allow auction participants to share a channel if they would interfere. Auctions which are not strategyproof include approaches that allocate power [17] and approaches that allocate each agent to his own channel [8, 12, 33]. VERITAS [35] was the first spectrum auction based on a monotone allocation rule. Zhou *et al.* [36] proposed TRUST, which uses a double auction for cases when multiple owners are selling channels. Jia *et al.* [23] envision spectrum owners auctioning off the right to use it as a secondary user when it is not otherwise being used by the owner and investigates an how revenue can be maximized in this setting. While winners share with the spectrum owner, there is no sharing among participants.

For auctions that permit sharing among auction participants, Gandhi *et al.* [13] use an approach that allocates many small channels, which effectively enables sharing based on demands of less than a full channel. However, their algorithm is not strategyproof and is not as expressive as ours. In particular, it allows sharing among bidders who might want only a portion of a channel but insists that their assignments do not overlap. Thus, it cannot take advantage of bidders who are not always active. Kasbekar and Sarkar [24] use a strategyproof auction and allow bidders to express arbitrary externalities, but they show that their approach is intractable except in a simple case.

We discuss the background of strategyproof auctions in general in Section 3. The issue of externalities in strategyproof auctions has been considered in a number of contexts. Jehiel *et al.* [21, 22] consider situations, such as the sale of nuclear weapons, where bidders care not just about winning but about who else wins. Krysta *et al.* [26] consider the problem of externalities in general combinatorial auctions. A number of papers have recently considered externalities in online advertising [15, 14, 25, 31].

3. CHALLENGES IN AUCTION DESIGN

In this section we describe the challenges that arise when designing a spectrum auction that permits sharing while being strategyproof and providing revenue to the auctioneer. First, we discuss the general form of an auction, introduce the notion of strategyproofness, and discuss why this is a desirable property. Second, we present a result by Myerson [29] from the economics literature, that provides a general framework for designing strategyproof auctions using a monotone allocation rule. Finally, we introduce the notion of a *reserve price*, a standard approach to increasing the revenue from an auction.

Auctions are a classic approach in economics to dividing goods among participants with competing needs and private values. In the simplest type of auction, a single item is sold to one of a number of bidders. Each bidder has private information about his value $V_i \in \mathbb{R}^+$. There are many ways such an auction can be run. One approach, known as a *first price auction*, is that each bidder names a price and the bidder who bids the most wins the item and pays what he bid. Another approach, due to Vickrey [34], is a *second price auction*, where each bidder names a price and the bidder who bids the most again wins the item. However, instead of paying the price he named, he pays the price named by the second highest bidder. We can adopt B_i (perhaps $\neq V_i$) to denote the bid submitted by i in an auction. Each bidder receives an allocation $A_i \in \{0, 1\}$, where $A_i = 1$ if the bidder gets the item and 0 otherwise. Feasibility would insist on $\sum_i A_i = 1$. Writing $B = (B_1, \ldots, B_n)$ for bids from n bidders, then we can write the allocation selected as a function $A(B) = (A_1(B), \ldots, A_n(B))$. Finally, each bidder makes some payment $P_i \in \mathbb{R}^+$ that depends on the bids, so we write $P_i(B)$. In a standard model, an bidder's utility, which captures his preference for the outcome of an auction, is

$$U_i(B) = V_i A_i(B) - P_i(B),$$

and represents his true value for the allocation minus the payment he makes.

Given these rules, how much should a bidder bid? In a first price auction, $P_i(B) = B_i$ for the winner, and so with perfect knowledge a bidder should bid slightly more than the highest bid of other bidders (to a maximum of V_i), since he wants to pay as little as possible. Thus bidders have an incentive to lie about their true value, and in doing so force other bidders to consider how they should respond to these lies (thus making the auction complex strategically). In contrast, in the second price auction, a bidder has a simple strategy that is (weakly) optimal no matter what other bidders do: bid his true value $B_i = V_i$. Such auctions, where it is optimal for a bidder to bid his true value, are known as *strategyproof. The key advantage of a strategyproof auction in our setting is this strategic simplicity*.

But how to design such a mechanism in our setting? One thing to recognize is that the allocation will be much more complicated: many channels are being allocated to many bidders, some of whom may ultimately share a channel. Part of the challenge will be describing a concise language to represent a bidder's value for different possible allocations. There is in fact a classic general auction design due to Vickrey, Clarke, and Groves [10, 16, 34] that is strategyproof. However, implementing this auction requires determining the optimal allocation of bidders to channels, which is an NP-Hard problem in our setting [20]. Another challenge is that the so-called VCG mechanism can have other bad economic properties in combinatorial settings [6].

If a heuristic solution is used instead, the resulting auction need not be strategyproof. However, as shown by Myerson [29] and introduced to the computer science literature by Archer and Tardos [5], if a monotone heuristic is used then prices can be computed that will make the auction strategyproof.

THEOREM 1. *An auction is strategyproof if and only if for all bidders i, and fixed bids of other bidders B_{-i},*

1. *$A_i(B)$ is a* **monotone** *function of B_i (increasing B_i does not decrease $A_i(B)$), and*
2. *$P_i(B) = B_i A_i(B) - \int_{z=0}^{B_i} A_i(z, B_{-i})\mathrm{d}z$.*

Hence, to achieve strategyproofness, monotonicity is of central importance in our approach. In the case of an auction for a single good, the nature of monotonicity is simple: either a bidder gets the good or not. In our case, this would be the equivalent of a bidder who demands a single channel either being assigned the channel or not. This is the approach used by VERITAS [35] and Jia *et al.* [23]. However, this is not sufficient due to the *externalities* which occur when an agent's value decreases because another agent is sharing the channel. Thus, our allocation rule must be monotone not only in whether a bidder gets a channel, but also how much sharing occurs on that channel.

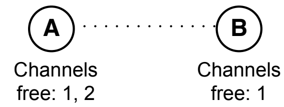


Figure 1: A potential violation of monotonicity. Nodes A and B are in contention range. At node A's location channels 1 and 2 are free; at B only channel 1 is free.

A first concern with auction design is to achieve *allocative efficiency*, allocating resources to maximize social welfare: the sum of bidder values. This is in contrast to much of the work in the systems community where the goal is to maximize throughput or spectral efficiency, both of which are not weighted by bidders' values. Thus, in addition to traditional metrics we also report social welfare in Section 5. Efficiency is often held to be of primary importance when designing a marketplace because it provides a competitive advantage over other markets and encourages participation by buyers. A second concern is to achieve a reasonable amount of revenue for the seller. This is relevant because it affects the incentive for a spectrum owner to participate. Paradoxically, auctions that manage to allocate more goods can sometimes raise less revenue. Suppose there are two agents participating in a second price auction and each wants a single item. If there is one unit of the item for sale, then the revenue will be the lesser of their values. However, if there are two units of the item, the revenue will be zero since each is guaranteed to get an item. A standard technique for increasing the revenue of an auction is to institute a minimum bid or *reserve price*. In Section 5.3, we add a reserve price to `SATYA` to enable a good trade-off between efficiency and revenue.

In summary, we would like `SATYA` to be strategyproof, which we achieve using a monotone allocation rule. We use a reserve price to increase revenue for spectrum owners.

4. THE `SATYA` ALGORITHM

4.1 Overview

As discussed in Section 3, the key to designing a strategyproof auction is to have an allocation rule that is monotone: the more an agent bids the greater his satisfaction with what he receives (given his true valuation). Prices can then be calculated that make the auction strategyproof. `SATYA` starts with a simple greedy algorithm that, when it considers an agent, allocates that agent to the best channel available (breaking ties by lowest channel number). We use two key techniques to make this greedy allocation monotone: *bucketing* and *ironing*.

To understand these techniques, it is important to understand how the greedy allocation can fail to be monotone. Throughout this section we adopt terminology *agent* to refer to an economic entity in the market. Figure 1 shows how an agent bidding more can result in him being less satisfied. If agent A has a lower bid than agent B, the algorithm assigns agent B to channel 1, then agent A to channel 2, and both are fully satisfied. If agent A raises his bid so that it is higher than agent B's bid, then the algorithm assigns him to channel 1. It has no other option than to assign agent B to channel 1, so the agents share and are less well off.

This example would be prevented if the algorithm was not allowed to assign agent B to channel 1 in the second case. We do this for many cases by assigning each agent to a "bucket" based on on his bid, such that the more an agent bids the higher the bucket to which he is assigned. Agents are not allowed to share with an agent from a higher bucket. Thus, in the example shown in Figure 1, if agent B is in a lower bucket than agent A, agent B will simply not be assigned a channel. If both agents are in the same bucket, we will consider them in some order independent of their actual bids, and adopt in place of their bid value the minimal possible value associated with the bucket. The effect is that the allocation decision is invariant to an agent's bid while the bid is in the same bucket. Since agents are only allowed to share with other agents within their buckets, the way buckets are chosen is an important parameter of our algorithm. Larger buckets create more possibilities for sharing. However, they also mean that the algorithm pays less attention to agent's bids, so they may decrease the social welfare (the total value of the allocation) and revenue.

Bucketing prevents many violations of monotonicity, but it is not sufficient to prevent all of them. In particular, the example from Figure 1 can still occur if agent A is in a lower bucket than agent B and then raises his bid so they are in the same bucket (if he raises it to be in a *higher* bucket there is no problem). To deal with this case we adapt a technique known as "ironing"[30] to this domain. This is a post-processing step in which allocations that might violate monotonicity are undone. For each agent allocated in the current bucket, we ask the counterfactual question "If this agent were instead in the next lower bucket, is it possible he would be allocated?" If so, we guarantee that the agent is satisfied in the current bucket by canceling (or "ironing") the allocations of other agents with whom he shares. In Figure 1, if agent A were in a lower bucket he would be allocated a channel. Therefore, in the ironing step, the algorithm would change agent B's allocation and not allocate a channel in the current bucket. It will be important, though, that a channel allocation that is cancelled in this way will be considered unavailable for future allocation. This prevents the need for nested arguments involving the effect of ironing on future allocations, future ironing of future allocations, and so on.

In this high level description, we have assumed that any two agents who interfere with each other cannot share a channel without harming each other. In reality, this is not the case; agents capable of using a MAC and sending at sufficiently low rates will have a negligible effect on each other. Many of the more intricate details of our algorithm come from adapting the general approach to take advantage of this fact and allow more efficient use of wireless spectrum.

4.2 Model

An auction mechanism for this problem takes the following as input:

- The number n of agents.
- A set of channels $\mathcal{C} \subseteq \mathbb{N} = \{1, 2, \ldots\}$. We denote the number of channels by $\chi = |\mathcal{C}|$.
- A conflict graph $G = (V, E)$. Each agent i is a vertex ($i \in V$). There is an edge $e = (i, j) \in E$ if the two agents would interfere with each other if they both broadcast on the same channel.
- A vector $C = (C_1, \ldots, C_n)$ that associates each agent i with the set $C_i \subset \mathcal{C}$ of channels that he can use.
- A vector $B = (B_1, \ldots, B_n)$ of bids that associates each agent i with his bid (per epoch) $B_i \in \mathbb{R}^+$. This bid is normalized for an agent's activiation probabilitiy, and (since our auction is strategyproof) represents the expected value to an agent for receiving as much of the share of a channel is it demands, without interference, in a single epoch.
- A finite set τ of agent types. Each type $T_i \in \tau$ is a four-tuple $T_i = (x_i, a_i, d_i, p_i)$ where
  - $x_i \in \{0, 1\}$ denotes whether an agent requires *exclusive use* of a channel in order to make use of it ($x_i = 1$) or is willing to share without another agent while active on the channel ($x_i = 0$). An exclusive use agent takes priority over any non exclusive use agent when both are active on the channel, while conflicting with another exclusive use agent that is active at the same time such that neither receives useful access. An exclusive use agent only requires access when active on the channel, and it may still make sense to try to allocate the channel to other agents for use when the agent is not active.
  - $a_i \in (0, 1]$ denotes the *activation probability* of an agent: how likely he is to actually want to use the channel at any given time. We think of channels as being used for a series of short epochs; a_i is the probability that the agent will be active in a given epoch and is assumed independent across epochs and across agents. For example, an agent may always want to use the channel ($a_i = 1$) or he might use it in a bursty fashion ($a_i = 0.1$).
  - $d_i \in (0, 1]$ is the *demand* of an agent; i.e., the portion of a channel that an agent who is willing to share ($x_i = 0$) demands in order to achieve full value when active. An agent's value falls off linearly for a share of the channel below d_i, and there is no additional

value for receiving a share of more than d_i. For example, an agent with $d_i = 0.5$ and bid value B_i would have an expected value of B_i when allocated 0.5 of the bandwidth on the channel whenever active and $0.4B_i$ when allocated 0.2 of the bandwidth whenever active (since $0.4 = 0.2/0.5$).

  - $p_i \in \mathbb{R}^+$ denotes the (per epoch) *penalty* an agent incurs if he wishes to use the channel and it is completely unavailable. The penalty amount is normalized for an agent's own activation probabilitiy, and represents the expected penalty incurred by an agent for having a completely unavailable channel whenever the agent wants to use the channel. A channel can only be unavailable in this sense when an exclusive use agent is also active and interfering on the channel; non exclusive use agents cannot will share rather than make a channel completely unavailable. Both exclusive use and non exclusive use agents can have a penalty – this can be incurred even for exclusive use agents when they are colocated with another exclusive use agent. The penalty represents, for example, the unhappiness of an ISP if it is unable to offer any connectivity for a period of time.

- A vector $T = (T_1, \ldots, T_n)$ that associates each agent i with his type T_i.

We assume throughout that only the bid value B_i is reported by an agent to the auction. This represents the agent's claim about its private value, which we denote V_i. An agent's type T_i and an agent's set of channels C_i is assumed to be known. In practice, these characteristics, such as how often the agent makes use of the channel, can be observed by the auctioneer and the agent can be punished if he lied.

To make our notion of a type concrete, consider the following examples of agents that might participate in such an auction. An agent who wishes to run a low-power (local) TV station on a channel would be unable to share it with others when active ($x_i = 1$), would be constantly broadcasting ($a_i = 1$), and would have a very large penalty p_i since it is unacceptable for the broadcast to be interrupted by someone turning on another (exclusive use) device. Another agent might want to use a device like a wireless microphone that also cannot share a channel when active ($x_i = 1$), but might be used only occasionally ($a_i = 0.05$) and might have a smaller value of p_i since it may be acceptable if it is occasionally unable to be used because there is another exclusive agent also trying to use the channel. For example, it might make sense to have several such devices share a channel if they interfere with each other sufficiently rarely.

There are also classes of agents capable of using a MAC and thus sharing a channel ($x_i = 0$). For example, someone who wants to run a wireless network could have constant traffic ($a_i = 1$) that consumes a large portion of the channel ($d_i = 0.9$), and might have a large penalty similar to a TV station because completely disconnecting users is unacceptable. However, such an agent is willing to share the channel with other non-exclusive types, and pay proportionately less for a smaller fraction of the bandwidth. There might also be opportunistic data users who occasionally ($a_i = 0.2$) would like to send a small amount information ($d_i = 0.4$) if the channel is available. Such users might have a low or even no penalty as their use is opportunistic.

Taken together, this information describes in a concise way the bid of an agent for all possible allocations. Based on bids B from agents, an auction mechanism outputs a vector $A = (A_1, \ldots, A_n)$ of assignments of agents to channels where $A_i \in C_i \cup \{\perp\}$ is the assignment to agent i. An assignment of $\perp$ indicates the agent has not been assigned a channel. Any assignment is feasible by definition, although an allocation in which many agents are allocated may be undesirable because it will generate lots of penalties due to exclusive use agents or very small fractional channel allocations amongst non exclusive use agents.

To determine an agent's bid value for an allocation, we need to reason about the kind of access an agent will receive to a channel. As we saw in our examples, some types of agents ($x_i = 1$) are unable to share a channel and preempt others who might be trying to use it. This is the case, for example, in whitespaces where new users must avoid sending if a TV station or wireless microphone is using the channel. To know how often such preemption occurs, and thus how often the agent experiences his penalty for being unable to access the channel, the agent cares how often the allocated channel is *free* (F) of exclusive agents. Given that no exclusive agent is using the channel, an agent capable of sharing ($x_i = 0$) might also end up sharing the channel with other agents. Thus, for example, an ISP also needs to know the expected share S of the bandwidth he gets for his users given that he is active and given that the channel is free.

We now formally define these two measures of channel availability. The first is the probability that the channel is free given an allocation A, i.e. the fraction of time no neighbor in the conflict graph that requires exclusive access is trying to use it. If N_i is the set of neighbors of i in G, we can write this as

$$\Pr_i(F|A) = \prod_{j \in N_i \ s.t. \ A_i = A_j \wedge x_j = 1} (1 - a_j).$$

This computes the joint probability that no neighbor allocated the same channel is active.

The second measure the agent cares about is, given that the channel is free and the agent is itself active, the expected fraction that is available to the agent to use. For an exclusive use agent this measure is just 1 because it gets complete access when the channel is free and it is active. But for non-exclusive agents we need to consider that the agent may be sharing with several neighbors. We first consider the effect of a fixed number of active neighbors. For example, if there are two other agents currently sharing a channel, we need to know how much of the channel's capacity each will get.

Agent i will not receive more than a d_i share: he doesn't have any more to send. We assume that agents use a Carrier Sense Multiple Access (CSMA) style MAC that shares bandwidth as evenly as possible amongst the active (non exclusive use) agents, subject to the constraint that no agent j receives more than its demand d_j. Formally, if N is the set of i's currently sending neighbors with whom he shares a channel and $N_f = \{j \in N \mid d_j < f\}$, he receives a share of the bandwidth equal to

$$\mathit{share}(N, i) = \min\left(d_i, \max_{f\in[0,1]} \frac{1 - \sum_{j\in N_f} d_j}{|N - N_f|}\right)$$

The agent either gets his full demand or, failing that, his fair share of the (which the $\max$ in the equation determines). This calculation is an approximation because it abstracts away the details of how the MAC actually manages contention. We discuss this issue further in Section 4.6.

But more than this, we also need to worry about a distribution on the number of neighbors that may be active in any given epoch. The value $\mathit{share}(N, i)$ is for some fixed set N of neighbors. Denote the total set of i's neighbors assigned to channel c according to A by

$$\nu(A, i, c) = \{j \in N_i \mid A_j = c\}.$$

Then the probability that a particular set $N \subseteq \nu(A, i, c)$ for $c = A_i$ (ν for brevity) is active is

$$\mathit{active}(N, \nu) = \left(\prod_{j\in N} a_j\right)\left(\prod_{\ell\in\nu-N}(1 - a_\ell)\right).$$

From this, an agent's expected share of the channel given that the agent is itself active and that the channel is free (where the expectation is computed over all possible activation patterns induced by allocation A), is then

$$E_A[S_i|F] = \begin{cases} 0 & \text{if } \Pr_i(F|A) = 0 \\ 1 & \text{if } x_i = 1 \\ \sum_{N\subseteq\nu} \mathit{active}(N, \nu) \cdot \mathit{share}(N, i) & \text{o.w.} \end{cases}$$

where "o.w." is "otherwise." For an agent for which the channel is never free (because a colocated exclusive user is always active) then we just say this is 0 but it is seen to be irrelevant in the final definition of expected value for an allocation. For an exclusive use agent, this is simply 1 because the agent prempts any sharing types when the channel is otherwise free of interference from exclusive use types. For the case of non-exclusive use agents, the expected share is computed in expectation over all possible subsets of neighbors that could be active. Note that this calculation is potentially quite expensive as it requires summing over all possible subsets of neighbors. We discuss this issue further in Section 4.5

We can now introduce the bid value that is implied by our bidding language for wireless auctions with sharing. An agent's expected bid value for an allocation A is

$$b(A, i) = \begin{cases} 0 \text{ if } A_i = \perp, \text{ otherwise} \\ B_i \Pr_i(F|A) E_A[S_i|F] - p_i(1 - \Pr_i(F|A)). \end{cases}$$

This combines his bid value for the amount of bandwidth he is actually able to use with the penalty for those times he is completely unable to access the channel.

Recall that B_i is the expected value to an agent given its own activation probability and when the channel is always free and a share d_i available when it is active. The first term is the expected value considering the probability the channel is in fact free and the expected share. We assume in constructing this expression a linear decrease in value to a non-exclusive use agent for an expected fraction below d_i (since the expected share received by an agent is $\Pr_i(F|A)E_A[S_i|F]$.)

Similarly, recall that p_i is the expected penalty to an agent given its own probability of activation (how much it would suffer on average over many epochs if the channel was always blocked when it was active.) Taken together, $b(A, i)$ represents the agent's *willingness to pay* for an allocation and the most an auctioneer could charge an agent. In practice, we charge less in order to achieve strategyproofness.

An agent also has a true per-epoch expected value V_i, and so the *true expected value* for an allocation is computed as,

$$u(A, i) = \begin{cases} 0 \text{ if } A_i = \perp, \text{ otherwise} \\ V_i \Pr_i(F|A) E_A[S_i|F] - p_i(1 - \Pr_i(F|A)). \end{cases}$$

4.3 Allocation Algorithm

Even if no agents are permitted to share channels, finding the optimal assignment of agents to channels is NP-Hard [20]. Therefore we adopt the same approach as previous strategyproof algorithms and make allocations greedily. To do so, our allocation algorithm assigns each agent i to a bucket K_i based on his bid B_i. There many ways this can be done as long as it is monotone in the agents bid. For example, agent i with a bid in the range $[2^\ell, 2^{\ell+1})$ could be assigned to bucket $K_i = \ell$. In general, we assume that this is done according to some function $\beta(k)$ such that bin k contains all agents with bids in the range $[\beta(k), \beta(k + 1))$.

The agents in each bucket are then assigned channels in descending order of buckets, with the order of assignment within a bucket determined randomly. Channels are assigned greedily from among the channels currently available to agent i. A c channel is *available* given allocation A if

- it is in C_i;
- it is not assigned to any j such that $(i, j) \in E$ and $K_j > K_i$ (a neighbor of i from a higher bucket) such that

$$\sum_{\ell\in\nu(A,j,c)\cup\{i,j\}} d_\ell > 1; \text{ and}$$

- The combined demands of i and his neighbors from higher buckets assigned to c are less than 1:

$$d_i + \sum_{j \in \nu(A,i,c) \cap \{\ell \mid K_\ell > K_i\}} d_j < 1$$

We refer to this condition as the demand of i being *satisfied*. Similarly, the second condition ensures that the demands of each neighbor from a higher bucket would be satisfied.

The second condition helps ensure a monotone allocation by preventing an agent from imposing any externality on an agent from a higher bucket (so that the agent in the higher bucket is unaffected by allocation decisions made in lower buckets). The third condition is also important for a monotone allocation to ensure that, in any higher bucket, i could have his demand satisfied.

For each available channel c and $\perp$, the algorithm calculates an estimate of the utility for *each* agent for the assignment A that results from assigning i to c, every currently unassigned agent to $\perp$, and leaving the other agents assigned as is as

$$e(A,j) = \beta(K_j) \Pr_j(F|A) E_A[S_j|F] - p_j(1 - \Pr_j(F|A))$$

Note that the estimate differs from the agent's actual bid by assuming that each agent in a given bucket shares the same value. Agent i is assigned to the channel that maximizes the sum of agent utilities (given the current assignment) while not giving any agent a negative utility. In the event of a tie, he is assigned to the lowest numbered among the tied channels, with $\perp$ being the lowest numbered channel.

After agents in a bucket are assigned channels, there is an "ironing" step. First, the allocation procedure is re-run for each agent to determine what would happen had he not been in his current bucket (or above). These counterfactuals are used to determine if the agent might have been able to be allocated a channel in a lower bucket. If so, as we show later, this might cause a monotonicity violation where an agent bids more but ends up less well off, so the provisional allocation is "ironed" by changing the assignments of the neighbors with whom he shared a channel to $\perp$.

This algorithm is specified in pseudocode as Algorithm 1. In the specification, we use distinct names to be able to refer to allocations created a long the way. The variable $A(k,i,j)$ denotes the state of the allocation in bucket k after considering the jth agent in the order given by the permutation π. Some of these allocations will be used for the conterfactual questions asked by ironing, so i is the agent currently being omitted ($i = 0$ if there is no such agent).

THEOREM 2. *Algorithm 1 is monotone. That is, for all agents i, changing bids to $B'_i > B_i$ changes the allocation from Allocation to Allocation′ such that $Pr_i(F|A) \geq \Pr_i(F|A)$ and $E_{A'}[S_i|F] \geq E_A[S_i|F]$.*

We defer the proof of Theorem 2 to the appendix.

Algorithm 1 Allocation Algorithm

```
π ← a random permutation of 1 ... n
M ← max_i K_i
m ← min_i K_i
Allocation_i ← ⊥∀i
A_i(M+1, 0, n) ← ⊥∀i
// Do Provisional Allocation
for k = M to m by −1 do
  A(k, 0, 0) ← A(k+1, 0, n)
  for j = 1 to n do
    A(k, 0, j) ← A(k, 0, j−1)
    if K_π(j) = k then
      c ← AssignChannel(A(k, 0, j), π(j))
      A_π(j)(k, 0, j) ← c
      Allocation_π(j) ← c
    end if
  end for
end for
// Counterfactuals to use for ironing
for i = 1 to n do
  A(K_i, i, 0) ← A(K_i + 1, 0, n)
  for j = 1 to n do
    A(K_i, i, j) ← A(K_i, i, j−1)
    if K_π(j) = K_i ∧ π(j) ≠ i then
      c ← AssignChannel(A(K_i, i, j), π(j))
      A_π(j)(K_i, i, j) ← c
    end if
  end for
end for
// Do ironing
for i = 1 to n do
  free ← ∃ avail. c for π(i) given A(K_π(i), π(i), n)
  if Allocation_π(i) ≠ ⊥ ∧ free then
    nbrs ← ν(Allocation, π(i), Allocation_π(i))
    while d_π(i) + Σ_{j∈nbrs} d_j > 1 do
      j ← last j ∈ nbrs according to π
      Allocation_j ← ⊥
      nbrs ← nbrs − {j}
    end while
  end if
end for
return Allocation
AssignChannel(A, i):
channels ← {c available for i given A}
for all c ∈ channels ∪ {⊥} do
  A_i ← c
  value_c = Σ_{j=1}^{n} e(A, j)
  if ∃j s.t. e(A, j) < 0 then
    value_c = 0
  end if
end for
return arg max_c value_c
(break ties in favor of ⊥, then lowest channel number)
```

4.4 Pricing Algorithm

Our pricing algorithm uses the insight from Theorem 1 to set prices so that the monotonicity of Algorithm 1 (Theorem 2) guarantees the auction is strategyproof. Note that the algorithm bases these (up front) payments on the expected value to the agent (based on the a_i), so his actual value might differ depending on the actual activation pattern of the other agents. In our case, these prices have a particularly simple form. Because of the way ironing works, there is exactly one bucket in which an agent can receive an allocation in which he shares a channel with other agents. In any lower bucket he does not get allocated a channel; in any higher bucket he is guaranteed by ironing to have his demand fully satisfied. Thus there are only three possible allocations and three possible prices. Algorithm 2 shows how this bucket can be determined and what price should be charged in each case.

Algorithm 2 Pricing Algorithm

$M \leftarrow \max_i K_i$
$m \leftarrow \min_i K_i$
for $i = 1$ to n **do**
  if $Allocation_i = \perp$ **then**
    $P_i = 0$
  else
    run Algorithm 1 without agent i to get $A'(k, 0, n) \forall k$
    $k = M$
    while $k > m - 1 \wedge \exists c \in C_i$
     s.t. $\nu(A'(k, 0, n), i, c) = \emptyset$ **do**
      $k = k - 1$
    end while
    // k is now the unique bucket in which i might share
    run Algorithm 1 with i in bucket k to get $Allocation'$.
    $f \leftarrow \Pr_i(F|Allocation')$
    $s \leftarrow E_{Allocation'}[S_i|F]$
    if $K_i > k$ **then**
      $P_i \leftarrow \beta(k+1) - (\beta(k+1) - \beta(k))fs$
    else
      $P_i \leftarrow \beta(k)fs - p_i(1-f)$
    end if
  end if
end for
return P

Recall that we assume that parameters such as an agents activation probability and demand are known to the auctioneer. Thus, when we say that the algorithm is strategyproof, we mean that agents have no incentive to lie about their *bids*. Since everything depends on this one value, our auction is an example of a *single-dimensional* auction (with externalities).

THEOREM 3. *The auction that allocates channels using Algorithm 1 and charges payments according to Algorithm 2 is strategyproof.*

Theorem 3 essentially follows from Theorem 1. However, the theorem would have to be slightly adapted because in our model agents' utilities depend on their penalty p_i in a way that is not linear in their valuation. Rather than proving a variant of Theorem 1, we simply use its approach to prove strategyproofness directly as the proof gives insight into how prices work in our case. We defer the proof to the appendix.

4.5 Running time of SATYA

Recall that n is the number of agents and χ is the number of channels. As we show, the running time of SATYA is determined largely by the implementation of the AssignChannel procedure. Unfortunately, calculating it for agent i requires time exponential in the number of neighbors with which i shares each channel considered. If sharing is limited to d neighbors in practice, this approach requires only time $O(\chi n 2^d)$. If d is small due to the nature of the conflict graph or an imposed constraint on the amount of sharing permitted, this approach is efficient enough. In our experiments we did not need to impose such a limitation.

THEOREM 4. SATYA*'s running time is determined by the time needed for $O(n^3)$ calls to AssignChannel.*

PROOF. SATYA needs to calculate $A(k, 0, n)$ for each non-empty bucket k and $A(K_i, i, n)$ for each agent i. There are at most n non-empty buckets and n agents for a total of $2n$ allocations to be computed. Each allocation requires assigning a channel to each agent at most once, so there are $O(n^2)$ calls to AssignChannel. Ironing takes time $O(\chi n)$ per agent for a total of $O(\chi n^2)$, so the running time of the allocation is dominated by the calls to AssignChannel (which needs at least time χ to consider each channel).

The pricing algorithm runs for each agent and runs the allocation algorithm twice: once to determine in which bucket the agent might share and once to determine what his share would be in that bucket. Thus SATYA requires $2n + 1$ allocations for a total of $O(n^3)$ calls to AssignChannel. □

4.6 Extensions

SATYA can be extended in a number of ways. In this section we discuss extensions to auctioning multiple channels, using other models for demand satisfaction, and the ability to customize aspects of the algorithm.

VERITAS [35] suggests a number of ways to extend channel auction algorithms to multiple channels. In particular, agents can either require a specific number of channels or be willing to accept a smaller number than they request. Agents may also wish to insist that their channel allocation be contiguous. SATYA can be extended to allow all of these. Due to space considerations we omit the details of the algorithmic changes required, but simulations of this extension are presented in Section 5.4.

As mentioned in Section 4.2, we use a simple model to calculate what happens when agents share a channel. Our simple model can be replaced my a more sophisticated model from prior work that has explored the capacity of CSMA

Agent Type	*Act. Prob.*	*Bid*	*Penalty*	*Demand*
Exclusive-Continuous	1	[0, 1000]	10000	1
Exclusive-Periodic	[0.05, 0.15]	[0, 1000]	5000	1
Sharing-High	1	[0, 1000]	10000	[0.3, 1]
Sharing-Low	[0, 1]	[0, 1000]	5000	[0.3, 1]

Table 1: Mix of agents used in the evaluation

based wireless networks (*e.g.,* [27, 37, 38, 28])as long as, in expectation, having more neighbors decreases an agent's share of the channel. This model can also be extended in other interesting ways. For example, we could add for each agent i a parameter ℓ_i such that if he receives less than an ℓ_i fraction of the channel it is useless. This simply requires defining his share to be 0 if it would be less than ℓ_i.

Finally, SATYA has a number of parameters that can be altered in various ways. On obvious choice is the function β used to assign agents to buckets. Any function that is monotone in an agent's bid can be used. This includes functions that take into account other facts about the agent, for example his type or the number of neighbors he has in the conflict graph. Another choice is the permutation π. Rather than choosing it randomly, any method that does not depend on agent bids can be used. Some natural possibilities include ordering agents by their degree in the conflict graph (so that agents who interfere less are allocated first), ordering by a combination of activation probability and demand (so that agents who use less spectrum are allocated first) or considering exclusive agents last since they impose much larger externalties on those with whom they share.

5. EVALUATION

In this section we compare the performance of SATYA to VERITAS. Since VERITAS does not permit sharing, we modified it slightly and implemented VERITAS-S, a variant of the algorithm that permits sharing as long as there are no externalities imposed. In VERITAS-S, an agent is assigned a channel only as long as it does not reduce the value of the agent or any of the neighboring nodes (in the conflict graph). We also implemented GREEDY, a version of SATYA without bucketing and ironing that provides higher overall efficiency. GREEDY is **neither** strategyproof nor monotone. Since it is not strategyproof, agents bids need not match their true values. However, to set as high a bar as possible, we assume they do so Since it gets to act on the same information but has fewer constraints than SATYA, GREEDY serves as an upper bound for our experiments.

Parameters: As shown in Table 1, all our experiments use four classes of agents bidding for spectrum. Each class represents different applications. For example, a TV station serving a local community is an agent who wants exclusive access for a long period of time. A wireless microphone is an example of an agent who wants exclusive access but for short periods of time. A low-cost rural ISP is an example of a Sharing-High agent who expects to actively use the spectrum but can potentially tolerate sharing, and a regular home user is an example of a Sharing-Low class agent whose spectrum access pattern varies. Note, each class of agents may have different transmit powers and coverage areas than the others. Since our goal is to evaluate the efficacy of SATYA in exploiting opportunities for sharing, we assign 5% of the total agents as exclusive-continuous, 15% exclusive-shared, 30% Sharing-High, and the remaining 50% Sharing-Low.

Methodology: Each auction algorithm takes as input a conflict graph for the agents. To generate this conflict graph in a realistic manner, we implement and use the popular Longley-Rice [2] propagation model in conjunction with high resolution terrain information from NASA [1]. This sophisticated model estimates signal propagation between any two points on the earth's surface factoring in terrain information, curvature of the earth, and climactic conditions. We use this model to predict the signal attenuation between agents, and consequently the conflcit graph between the bidding agents.

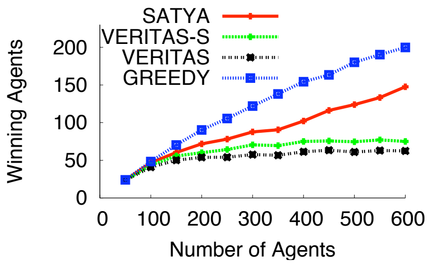
We use the FCC's publicly available CDBS [11] database to model the transmit power, location, and coverage area of Exclusive-Continuous users. Note, this information as well as the signal propagation predictions are sensitive to geographic areas. We model the presence of all other types of agents using population density information.

Agents are scattered across a 25 mile x 25 mile urban area in a random fashion by factoring in population density information. Since each class of agent has a different coverage area, depending on the type of agent, we determine a pair of conflicting nodes if the propagation model predicts signal reception higher than a specified threshold. We repeat each run of the experiment 10 times and present averaged numbers across runs. Unless otherwise specified, the number of channels is 5 and, except for Section 5.4, each agent only desires a single channel. In tuning SATYA, we experimented with a variety of methods for determining to which bucket to assign an agent. We do not present these results for space reasons, but based on them use buckets of size 500 ($\beta(k) = 500k$).

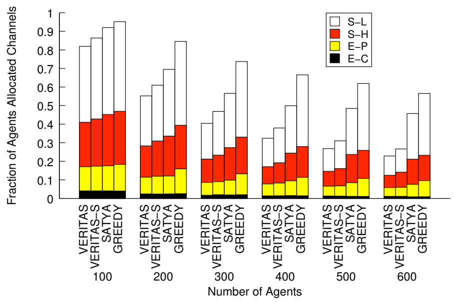
In our experiments, we us the following metrics to evaluate SATYA, VERITAS, VERITAS-S, and GREEDY.

- *Winning Agents*: The total number of agents that are allocated at least one channel by the auction algorithm.
- *Social Welfare*: The sum of the valuations for the allocatioion by winning agents (includes externalities).
- *Satisfaction:* The sum of the fraction of his demand each agent had satisfied.
- *Spectrum Utilization*: The sum of satisfiaction weighted by activation probability and demand. channel each winning agent receives by the auction algorithm.
- *Revenue*: The sum of agents' payments.

Social welfare is a generally accepted measure in economics of the total happiness of agents. From a networking perspective, spectrum utilization is a measure of how much the spectrum is being used (similar to the total network capacity). Agents allocated and satisfaction are measures of



(a) Winning Agents



(b) Distribution of Winning Agents

Figure 2: Number and Type of Winning Agents

how many and to what extent the algorithm is able to satisfy agent demands. Revenue is how much spectrum owners benefit, and thus their incentive to participate.

5.1 Varying the Number of Agents

Figure 2 shows the performance of various algorithms as a function of the number of agents participating in the auction. As we vary the number of agents, we keep the mix of bidders to be the same as Table 1.

As seen in Figure 2(a), as the number of agents increases, `SATYA` produces up to 72% more winning agents when compared to VERITAS and VERITAS-S. This gain comes from being permitted to allocate agents despite externalities. With fewer agents, all three algorithms demonstrate similar performance because almost all agents can either be allocated a channel of their own or are impossible to satisfy.

Overall, VERITAS-S and VERITAS do not make the best use of possible bidders who can share. This is demonstrated in Figure 2(b), which is the distribution of different classes of bidders assigned channels by each algorithm. As the number of agents increases, VERITAS-S and VERITAS significantly reduce the fraction of agents capable of sharing who are assigned channels (relative to `SATYA`). However, all algorithms demonstrate a similar performance in the fraction of exclusive bidders who are assigned channels. Hence, `SATYA` is capable of taking advantage of sharing by allocating channels to more of such users. As expected GREEDY, which makes no effort to avoid externalities that would violate monotonicity, outperforms all strategyproof auctions and is able to assign more sharing agents. Although we omit the data for space, the difference in performance between `SATYA` and GREEDY is primarily due to bucketing. Ironing does occur but has only a minor effect.

In addition to the number of agents allocated spectrum, the results for other metrics are shown in Figure 3. As seen in Figure 3(a), the total social welfare attained by `SATYA` increases with an increase in the number of agents. This is a direct consequence of assigning channels to more agents capable of sharing the spectrum. This shows that, despite externalities from sharing, the additional agents allocated consider it valuable. At 600 bidders, `SATYA` realizes a gain of 25% over VERITAS-S and 40% over VERITAS in the total social welfare of the network. Similarly, as seen in Figure 3(b), we find a 50-80% increase in the spectrum utilization of the network using `SATYA`. As social welfare, spectrum utilization and satisfaction all take into account externalities, Figures 3(a), 3(c), and 3(b) show significant correlation. As with the agents allocated metric, at fewer nodes the algorithsms are essentially indistinguishable as there are few opportunities to share.

Hence, the main takeaway is that, *by using the bucketing mechanism,* `SATYA` *increases the number of winning agents as well as the total utility and use of the assigned spectrum.*

5.2 Varying the Number of Channels

We also measured the effect of varying the number of channels auctioned on the overall outcome of the auction. The results shown in Figure 4 demonstrate the following trend: as the number of auctioned channels increases the gap in performance among the algorithms reduces. This is similar to having fewer bidders participate in the auction; with more channels, there is a reduced need for sharing and all algorithms perform similarly. As Figure 4(a) shows, `SATYA` is still able to assign more bidders than other algorithms until about 25 auctioned channels. Similarly, in Figure 4(b), we see that `SATYA` outperforms the other algorithms by 20-60% in social welfare up until about 10 channels.[1]

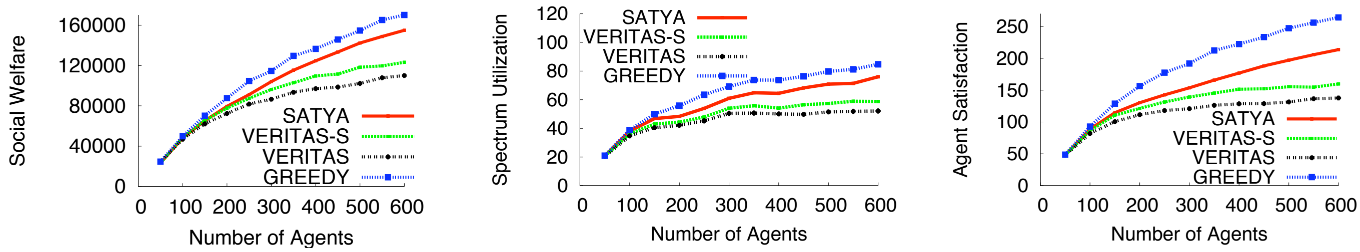
We also varied the number of agents and the number of channels simultaneously and the results for `SATYA` are shown in Figure 4(c). We see that as the number of agents increases, `SATYA` takes advantage of the increased opportunity for sharing and allocate more agents.

Hence, the main takeaway is `SATYA` *provides substantial benefits for those scenarios where the number of channels makes spectrum scarce.*

5.3 Measuring Revenue

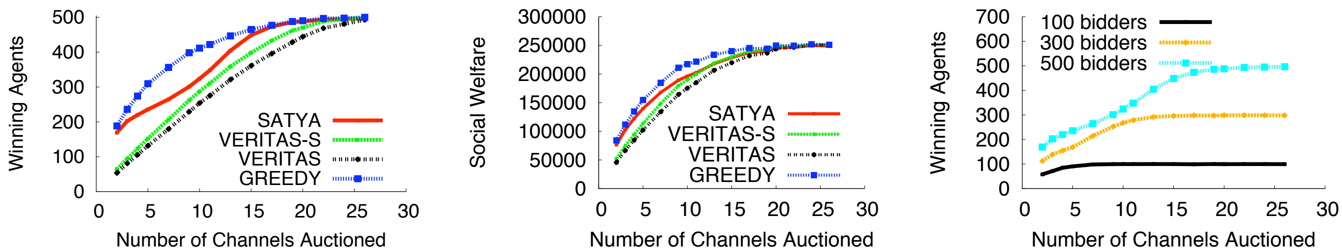
We consider social welfare the most important measure of performance: a market that finds success in the long run will allocate resources to those that find the most value. However, in our setting revenue may also be important to provide an incentive for current spectrum owners to participate in the secondary market. The pricing algorithm presented in Section 4.4 determines the total revenue obtained by the auctioneer. First, we measure the total revenue obtained as a

[1]We elide graphs for spectrum utilization and satisfaction for this and remaining experiments due to a lack of space; they demonstrate a similar trend.



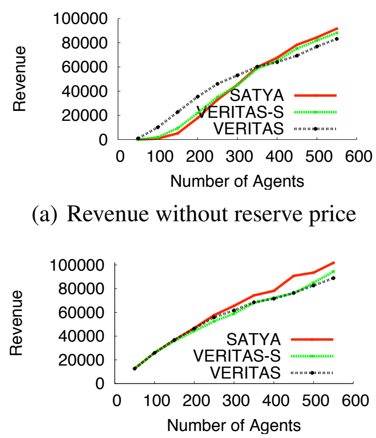
(a) Social Welfare (b) Spectrum Utilization (c) Satisfaction

Figure 3: Effect of varying the number of agents in the auction



(a) Winning Agents (b) Social Welfare (c) Winning Agents for `SATYA`

Figure 4: Effect of varying the number of channels auctioned



(a) Revenue without reserve price

(b) Revenue with reserve price

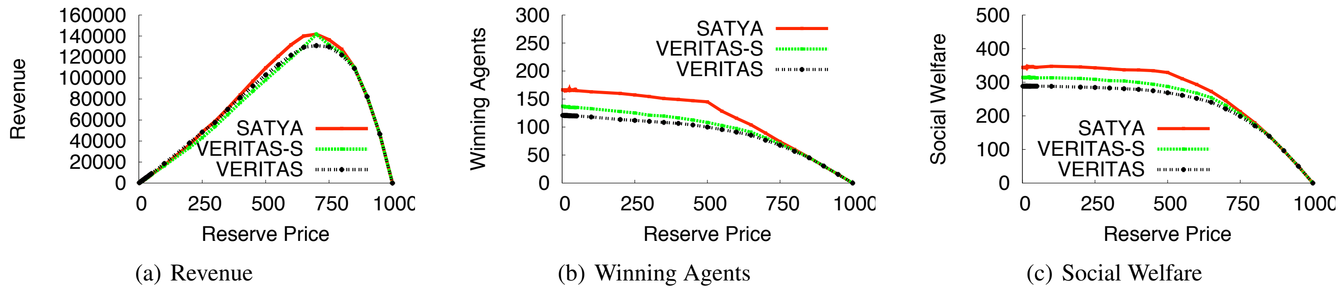
Figure 5: Revenue as a function of number of agents.

function of the number of agents bidding for spectrum *without* reserve prices. We do not include GREEDY in this analysis because it is not strategyproof and it is not clear what agents will bid and thus what the actual revenue would be. As seen in Figure 5(a), the revenue obtained by `SATYA` and VERITAS-S is much lower than VERITAS for smaller numbers of agents. As explained in Section 3, this is a direct consequence of sharing making it easier to accomodate agents: if they will be allocated with a bid of zero they do not have to pay anything in a strategyproof auction.

To improve revenue in such situations, we use a reserve price, a technique applicable to many auction designs. The basic idea is that there is a minimum bid an agent must make to participate in the auction. This does not affect strategyproofness and is implemented by dropping lower bids and modifying the pricing algorithm to value a channel as worth at least the minimum bid. VERITAS explored a similar opportunity to increase revenye by limiting the number of channels available. To make best use of a reserve price, the auctioneer needs a good way to determine what this minimum bid should be. In simple situations, the optimum value can be determined theoretically [29]. However, our model is sufficiently complex that we determine a value empirically.

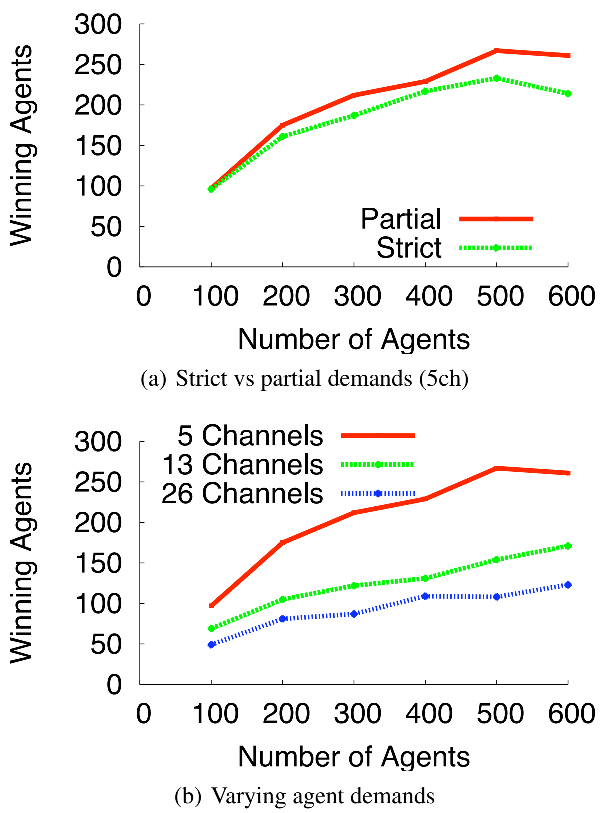
The results from a simulation that varies the reserve prices is shown in Figure 6 for 300 bidding agents. As seen in Figure 6(a), with a reserve price of 0 (i.e. no reserve price), VERITAS performs better than `SATYA` and VERITAS-S. However, as the reserve price begins to increase, the revenue derived from all three auctions increases. This is because we are able to allocate nodes while charging them a reserve price if they fall into the lowest bucket. However, at around a price around 700 (depending on the algorithm), there is an inflection point in the revenue. As seen in Figure 6(b), this is because significantly fewer agents are allocated by the auction and social welfare decreases (Figure 6(c)). Hence, reserve prices must *balance the revenue increase against the loss of social welfare*.

Based on these results, we use a reserve price of 400 and repeat the experiment to measure revenue by varying the number of bidders. We used a fixed reserve price for consistency; in practice it could depend on the number of agents. As Figure 5(b) shows, the revenues for the auctioneer increase significantly for all algorithms relative to Figure 5(a). This is most pronounced with 50 agents where revenue goes



(a) Revenue (b) Winning Agents (c) Social Welfare

Figure 6: Effect of reserve prices with 300 agents



(a) Strict vs partial demands (5ch)

(b) Varying agent demands

Figure 7: Experiments with multiple channels

from essentially zero to approximately ten thousand. SATYA, which without a reserve price lost revenue by being too efficient in allocating agents, benefits slightly more than the other two algorithms. With a large number of agents, the reserve price is essentially irrelevant because of the amount of competition; with 550 agents the gain is below **XX**%.

5.4 Multiple Channels

SATYA is also capable of allocating multiple channels when agents bid for multiple channels in an auction. To illustrate this, we ran an experiment where we varied the number of channels that each agent bids for as well as the number of agents in the auction. Not all agents bid for the same number of channels. The number of bid for is what an agent with $d_i = 1$ would request; lower d_i results in a proportionally lower request. We used two modes of channel allocation schemes in SATYA, *strict*: when an agent either gets the number of channels it requests for or nothing, and *partial:* an agent can get fewer than requested channels. The total number of channels auctioned (not to be confused with the number of channels bid) was fixed to 26. The results are shown in Figure 7. As seen in Figure 7(a), partial allocations result in slightly higher winning agents than strict, which is what we would expect since strict allocations are constraints that are harder to satisfy. Figure 7(b) shows that increasing the the number of channels demanded by agents reduces the number of winners as would be expected.

6. CONCLUSIONS

We have presented SATYA, the first tractable, strategyproof auction algorithm that deals with externalities and enables sharing of wireless spectrum among heterogeneous users. This is achieved by using "bucketing" and "ironing" to achieve a monotone allocation rule and charging appropriate prices. Using realistic simulations, we showed that the ability of SATYA to share channels results in superior allocations by a variety of metrics. We also showed that revenue can be increased using reserve prices, which provides spectrum owners with a stronger incentive to participate.

7. REFERENCES

[1] Shuttle Radar Topograph Mission (SRTM), http://www2.jpl.nasa.gov/srtm/.

[2] The ITS Irregular Terrain Model Algorithm, NTIA, Department of Commerce.

[3] OET BULLETIN No. 69, Longley-Rice Methodology for Evaluating TV Coverage and Interference. February 2004.

[4] FCC press release, Evaluation of the Performance of Prototype TV-Band White Space Devices. November 2008.

[5] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[6] L. M. Ausubel and P. Milgrom. The lovely but lonely Vickrey auction. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, chapter 1, pages 17–40. MIT Press, 2006.

[7] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. DIMSUMNet: new directions in wireless networking using coordinated dynamic spectrum access. In *2005 International Conference on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, pages 78–85, 2005.

[8] M. M. Buddhikot and K. Ryan. Spectrum management in coordinated dynamic spectrum access networks. In *Proc. of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks(DySPAN)*, 2005.

[9] M. M. Bykowsky, M. A. Olson, and W. W. Sharkey. Modeling the efficiency of spectrum designated to licensed service and unlicensed operations. Technical report, FCC OSP Working Paper Series, 2008.

[10] E. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[11] FCC Media Bureau. TVQ TV Database, http://www.fcc.gov/mb/video/tvq.html.

[12] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri. A general framework for wireless spectrum auctions. In *Proc. of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks(DySPAN)*, 2007.

[13] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri. Towards real-time dynamic spectrum auctions. *Computer Networks*, 52(4):879–897, 2008.

[14] A. Ghosh and M. Mahdian. Externalities in online advertising. In *17th International World Wide Web Conference (WWW)*, 2008.

[15] R. Gomes, N. Immorlica, and E. Markakis. Externalities in keyword auctions: An empirical and theoretical assessment. In *Internet and Network Economics, 5th International Workshop (WINE)*, pages 172–183, 2009.

[16] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[17] J. Huang, R. A. Berry, and M. L. Honig. Auction mechanisms for distributed spectrum sharing. In *Proc. of 42nd Allerton Conference*, 2004.

[18] S. B. Inc. Analysis of fcc secondary spectrum markets positions, policies and comments.

[19] S. B. Inc. The secondary spectrum market: A licensing & leasing primer.

[20] K. Jain, J. Padhye, V. Padmanabhan, and L. Qui. Impact of Interference on Multi-hop Wireless Network Performance. In *MOBICOM*, 2003.

[21] P. Jehiel, B. Moldovanu, and E. Stacchetti. How (not) to sell nuclear weapons. *The American Economic Review*, 84(6):814–829, Sep. 1996.

[22] P. Jehiel, B. Moldovanu, and E. Stacchetti. Multidimensional mechanism design for auctions with externalities. *Journal of Economic Theory*, 85:258–293, 1999.

[23] J. Jia, Q. Zhang, Q. Zhang, and M. Liu. Revenue generation for truthful spectrum auction in dynamic spectrum access. In *Proceedings of the 10th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 3–12, 2009.

[24] G. S. Kasbekar and S. Sarkar. Spectrum auction framework for access allocation in cognitive radio networks. In *Proceedings of the 10th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 13–22, 2009.

[25] D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. In *Internet and Network Economics, 4th International Workshop (WINE)*, pages 585–596, 2008.

[26] P. Krysta, T. Michalak, T. Sandholm, and M. Wooldridge. Combinatorial auctions with externalities. In *Ninth International Conference on Autonomous Agents And Multiagent Systems (AAMAS)*, 2010.

[27] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *MOBICOM*, 2001.

[28] V. Mhatre and K. Papagiannaki. Optimal Design of High Density 802.11 WLANs. In *CoNEXT*, 2006.

[29] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[30] D. C. Parkes and Q. Duong. An ironing-based approach to adaptive online mechanism design in single-valued domains. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 94–101, 2007.

[31] D. Reiley, S.-M. Li, and R. Lewis. Northern exposure: A field experiment measuring externalities between search advertisements. In *Proc. of 11th ACM Conference on Electronic Commerce (EC)*, 2010.

[32] J. S. Rosenschein and G. Zlotkin. Consenting agents: Designing conventions for automated negotiation. *AI Magazine*, 15(3):29–46, Fall 1994. Also reprinted in *Readings in Agents*, edited by Michael N. Huhns and Munindar P. Singh, Morgan Kaufmann Publishers, San Francisco, California, 1997, pages 353–370.

[33] A. P. Subramanian, M. Al-Ayyoub, H. Gupta, S. R. Das, and M. M. Buddhikot. Near optimal dynamic spectrum allocation in cellular networks. In *Proc. of 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks(DySPAN)*, 2008.

[34] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[35] X. Zhou, S. Gandhi, S. Suri, and H. Zheng. ebay in the sky: strategy-proof wireless spectrum auctions. In

Proceedings of the 14th Annual International Conference on Mobile Computing and Networking (MOBICOM), pages 2–13, 2008.

[36] X. Zhou and H. Zheng. TRUST: A general framework for truthful double spectrum auctions. In *INFOCOM 2009. 28th IEEE International Conference on Computer Communications*, pages 999–1007, 2009.

[37] J. Zhu, X. Guo, S. Roy, and K. Papagiannaki. CSMA Self-Adaptation based on Interferene Differentiation. In *Globecom*, 2007.

[38] J. Zhu, B. Metzler, Y. Liu, and X. Guo. Adaptive CSMA for Scalable Network Capacity in High-Density WLAN: a Hardware Prototyping Approach. In *Infocom*, 2006.

Appendix

Proof of Theorem 2

First, we observe that an agent's bid is only used to determine his bucket and is afterward ignored by the algorithm (estimates of utility use the agent's bucket rather than his bid). Thus is it sufficient to consider deviations that cause i to change buckets. If $A_i = \perp$, then $\mathrm{Pr}_i(F|A) = E_A[S_i|F] = 0$, so the claim is trivially true. Otherwise, i moves up to some bucket $k_2 > k_1$. Recall that $\nu(A, i, c) = \{j \in N_i \mid A_j = c\}$ denotes the set of i's neighbors assigned to channel c according to A. An important observation about the algorithm is that once it makes an assignment that some $A_i(k, 0, j) = c$, it never changes this for any later k and j. This is the reason the ironing step only changes $Allocation$ and not A. Thus, the set $\nu(A(k, 0, j), i, c)$ grows monotonically as the algorithms iterates over k and j.

Since i was assigned to c in the assignment A, c must have been available to him when he was assigned. By the third part of the definition of availability and the monotonic growth of ν, i would have his demand satisfied with neighbors $\nu(A(k, 0, j), i, c)$ for all $k \geq k_1 + 1$ and all j. In particular, this means his demand is satisfied with neighbors $\nu(A(k_2, 0, \pi^{-1}(i) - 1), i, c)$.

When computing $Allocation'$ with the new bids B', the algorithm computes a new set of incremental allocations A'. Since the algorithm does not look ahead, $A'(k_2, 0, \pi^{-1}(i) - 1) = A(k_2, 0, \pi^{-1}(i) - 1)$. This means that, in AssignChannel($A'(k_2, 0, \pi^{-1}(i)), i$)), i could be assigned to c and have his demand satisfied. Therefore he will be assigned to some such channel c' (not necessarily c as there might be a lower numbered channel available). Furthermore, on c he does not impose any externality on his neighbors (all their demands are satisfied by the second part of the definition of availability). Therefore, since the algorithm greedily maximizes the total value, this true on c' as well.

Again since the algorithm does not look ahead, i increasing his bid does not change anything before bucket k_2, so $A'(k_2 + 1, 0, n) = A(k_2 + 1, 0, n)$. Since the algorithm does not consider allocating i a channel in bucket k_2 when computing A (because he is in the lower bucket k_1) or when asking the counterfactual about what would have happened had i not been in bucket k_2 in A', $A'(k_2, i, n) = A(k_2, 0, n)$. Thus $\nu(A'(k_2, i, n), i, c) = \nu(A(k_2, 0, n), i, c)$ and in the ironing step running on B'_i,all of i's neighbors with which it might have shared a channel with be reassigned to $\perp$ until its demand is satisfied. Since $i's$ neighbors were satisfied when i was assigned to c' and neighbors are ironed in the opposite order from that in which they were added, i will not be ironed by any of its neighbors. Thus $Pr_i(F|A) = E_{A'}[S_i|F] = 1$ and the allocation is monotone.

Proof of Theorem 3

As observed there are only 3 possible allocations and sets of prices. An agent either gets nothing and pays nothing for a utility of 0, ends up in bucket k in which he might share and gets $V_i fs - p_i(1-f)$ and pays $\beta(k) fs - p_i(1-f)$ for a utility of $(V_i - \beta(k)) fs$, or ends up in a higher bucket and gets V_i (he has a channel to himself) and pays $\beta(k+1) - (\beta(k+1) - \beta(k)) fs$ for a utility of $V_i - \beta(k+1) + (\beta(k+1) - \beta(k)) fs$.

First suppose that $V_i < \beta(k)$. If he ends up sharing his utility is $(V_i - \beta(k)) fs < 0$. If he ends up with a channel to himself his utility is

$$V_i - \beta(k+1) + (\beta(k+1) - \beta(k)) fs < (\beta(k+1) - \beta(k))(fs - 1) < 0.$$

Thus his optimal strategy is to bid his true value and get $\perp$.

Now suppose that $\beta(k) \leq V_i \leq \beta(k+1)$. If he bids truthfully, his utility is $(V_i - \beta(k)) fs \geq 0$, so he cannot gain by lowering his bid. If he raises his bid above $\beta(k+1)$ he will end up with a utility of

$$\begin{aligned} V_i - \beta(k+1) + (\beta(k+1) - \beta(k)) fs &= \\ (V_i - \beta(k+1))(1 - fs) + (V_i - \beta(k)) fs &\leq \\ (V_i - \beta(k)) fs. \end{aligned}$$

Thus his optimal strategy is to bid his true value and share.

Finally, suppose that $V_i > \beta(k+1)$. If he bids truthfully, his utility is $V_i - \beta(k+1) + (\beta(k+1) - \beta(k)) fs > 0$, so he does better than if he is not allocated. If he lowers his bid to be in bucket k, his utility is

$$(V_i - \beta(k)) fs \leq V_i - \beta(k+1) + (\beta(k+1) - \beta(k)) fs.$$

Thus his optimal strategy is to bid his true value.